Rule CIC163: The number of buffers is too low for VSAM LSR pool

Finding: CPExpert has detected that file access requests to a Local Shared Resources (LSR) pool were required to wait for buffers. However, the CICS statistics did not reveal that storage was a constraint to CICS performance.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: A VSAM buffer is used to hold each VSAM control interval (CI) required to respond to a file access request. The number of data and index buffers are explicitly specified for Nonshared Resources (NSR) files. However, LSR files share a common pool of buffers, and there is no preallocation of buffers to particular files, to data or index, or to strings. Additionally, the LSR pool of buffers may be divided into "subpools" where each subpool consists of buffers of a specific size.

For LSR VSAM files, the number of buffers of each size can be specified explicitly (using the BUFFERS operand in the DFHFCT TYPE=SHRCTL macro or the DEFINE LSRPOOL command of Resource Definition Online).

Alternatively, CICS can automatically compute the number of buffers for the LSR pool, based upon the characteristics of files assigned to the pool. There are advantages and disadvantages to allowing CICS to compute the number of buffers in the LSR pool.

- It is easier to allow CICS to perform the computations. Additionally, if the number of buffers in the LSR pool is explicitly defined, the definition usually should be altered when additional files are assigned to the LSR pool or when files are removed from the pool.

- However, allowing CICS to compute the number of buffers for the LSR pool requires additional overhead (at startup) to build the LSR pool. This is because CICS must read the VSAM catalog for each file assigned to the LSR pool.

  Additionally, if CICS is allowed to compute the number of buffers in the LSR pool, there is no ability to explicitly specify the number of buffers for performance improvement.

It is generally better to explicitly specify the number of buffers assigned to the LSR pool. The decreased flexibility of allowing CICS to compute the number of buffers often outweighs the savings in programmer time required to make the specifications.

The CICS statistics provide information about the number of requests waiting for a buffer at any one time, and the total number of times request had to wait for a buffer. This information is provided by file name, and information is given about the size of the data and index buffers associated with the file.

Unfortunately, the information provided does not reveal whether the wait was for data or index buffers. Since the CI size is generally different for data and index, the information does not indicate which LSR subpool buffers were unavailable. However, CICS performance is constrained when any file access operations must wait for buffers.

CPExpert produces Rule CIC163 if the CICS statistics revealed that any files waited for buffers in an LSR pool. CPExpert provides information regarding the data buffer size and index buffer size associated with the file experiencing the wait for buffers.

Suggestion: CPExpert suggests that you consider the following:

- If you are explicitly defining the number of buffers assigned to this LSR pool, examine the DFHFCT TYPE=SHRCTL,BUFFER operand for the LSR pool. The BUFFER operand shows the CI size for the data and index buffers, and the number of buffers allocated to each size. You should increase the number of buffers for the size or sizes associated with the file.

  As mentioned above, the CICS statistics do not reveal which LSR subpool caused the waits. The wait on buffer could be for the LSR subpool of the data buffer size or for the LSR subpool of the index buffer size.

  However, the statistics do reveal the number of buffer reads and writes, by LSR subpool. CPExpert suggests that you increase the buffers of the LSR subpool with the largest number of buffer reads and writes. It is more likely that buffers of that size would be in short supply.

  The above conclusion is not certain, however. The wait on buffers could occur at relatively short intervals of high activity for a particular LSR subpool, even though the LSR subpool could have low **overall** activity.

Unfortunately, the approach must be to increase the buffer allocation for a LSR subpool and see whether that increase caused the wait on buffers problem to vanish. If the problem did not vanish, then the problem likely is related to the other LSR subpool.

- If you are allowing CICS to compute the number of buffers assigned to this LSR pool, you should increase the value of the RSCLMT operand in the SHRCTL macro.

  After CICS computes the total number of buffers of each size required by all files assigned to the LSR pool, CICS reduces this number by 50% or to the percentage specified in the RSCLMT operand (the RSCLMT operand value takes precedence). CICS makes sure that there is at least three buffers in each LSR subpool).

  If you previously specified a value for the RSCLMT operand, the value should be increased. If you did not previously specify a value for the RSCLMT operand, specify a value higher than 50 for the operand. The value should be increased until there are no instances in which files wait on buffers.

  This method is imprecise and applies to all buffers in the LSR pool. The limitations of the method illustrate another advantage of explicitly specifying LSR pool operands, rather than allowing CICS to compute the operands.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68 and pages 232-238.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 173, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73 and pages 93-106.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152 and page 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162 and page 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2 and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2 and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.1 and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.